Computer Science Review 43 (2022) 100450

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review article

Secure and privacy-preserving crowdsensing using smart contracts: Issues and solutions

Alfredo J. Perez [a,*], Sherali Zeadally [b]

[a] TSYS School of Computer Science, Columbus State University, 4225 University Ave, Columbus, GA 31907, USA
[b] 315 Little Library Building, College of Communication and Information, University of Kentucky, Lexington, KY 40506-0224, USA

ARTICLE INFO

Article history:
Received 14 July 2021
Received in revised form 3 November 2021
Accepted 20 November 2021
Available online xxxx

Keywords:
Blockchain
Crowdsensing
Internet of Things (IoT)
Privacy
Security
Smart contracts

ABSTRACT

The advent of Blockchain and smart contracts is empowering many technologies and systems to automate commerce and facilitate the exchange, tracking and the provision of goods, data and services in a reliable and auditable way. Crowdsensing systems is one type of systems that have been receiving a lot of attention in the past few years. In crowdsensing systems consumer devices such as mobile phones and Internet of Things devices are used to deploy wide-scale sensor networks. We identify some of the major security and privacy issues associated with the development of crowdsensing systems based on smart contracts and Blockchain. We also explore possible solutions that can address major security concerns with these systems.

Contents

1. Introduction

The high number of sensor-enabled Internet connected devices such as smartphones and Internet of Things (IoT) devices

* Corresponding author.
E-mail addresses: perez_alfredo@columbusstate.edu (A.J. Perez), szeadally@uky.edu (S. Zeadally).

https://doi.org/10.1016/j.cosrev.2021.100450
1574-0137/

are enabling new kinds of business ventures and societal applications that are exploiting these devices not only for profit but also for the benefit of the public. As of summer of 2020, and according to recent research (as of 2021) there are around 8 billion mobile subscriptions in the world, with 5.5 billion being smartphone subscriptions [1]. These numbers are expected to soar in upcoming years as technologies such as 5G/6G networks and more IoT devices are deployed around the world. In the past, we have seen applications of crowdsensing systems in areas such as environmental monitoring, transportation, entertainment, security, and healthcare [2]. More recently, many countries have deployed crowdsensing systems in response to the Coronavirus Disease 2019 (COVID-19) pandemic not only for epidemiology reasons (i.e., contact tracing [3,4]), but also for treatment [5].

In addition to crowdsensing, a second set of technologies are having a tremendous impact on society. These technologies are Blockchain and smart contracts. Blockchain offers several services for secure data storage, retrieval, and sharing with properties such as immutability, transparency, decentralization, and fault tolerance [6]. Smart contracts expand Blockchain technology by providing means to automate transactions in a Blockchain system through the specification of computer programs that encapsulate business logic and code needed to execute some actions when conditions are met [7]. Smart contracts enable crowdsensing to improve not only data collection and sharing in crowdsensing systems, but also to create opportunities in the development of decentralized markets wherein sensor data collectors can sell their data without the need of a centralized entity or a broker [8, 9]. However, this vision exposes various security issues that must be addressed. In this work, we explore these emerging issues along with possible solutions.

Research contributions of this work

We summarize the main research contributions of this work as follows:

- We present a review of crowdsensing and smart contracts.
- We explore security and privacy issues when enhancing crowdsensing with smart contracts and we present solutions that can address the security and privacy issues identified.
- We discuss open challenges that must be addressed in the future to enable the implementation of crowdsensing systems with smart contracts.

The rest of this paper is organized as follows. Section 2 presents a review crowdsensing systems and smart contracts. In Section 3 we explore security issues associated with the development of crowdsensing systems with smart contracts. Section 4 presents possible solutions to address these issues. In Section 5, we discuss some open issues that still need to be addressed for crowdsensing systems with smart contracts in the future. Section 6 presents concluding remarks and future work.

2. Crowdsensing and smart contracts

2.1. Crowdsensing systems

The history of modern research in Wireless Sensor Networks (WSN) started with the Distributed Sensor Networks (DSN) program developed in the 1970's in the United States [10]. This project used minicomputers and acoustic sensors to develop a system that could track low-flying aircrafts and it was considered state-of-the art during its time. The DSN project paved the way for a revolution in WSN technology and systems in the late 90's, in which networks of potentially thousands of small devices left unattended and interconnected wirelessly could monitor large areas of interest for many months, potentially years. However, actual implementations of WSNs were small-scale systems, with local and specialized focus because of deployment and maintenance costs which have made WSNs with thousands of devices impractical [11].

During the first and second decade of the 21st century, crowdsensing systems have emerged to alleviate the deployment and maintenance costs incurred in the massive use of single WSN systems with thousands of devices by leveraging the utilization of billions of smartphones and other IoT devices owned by the general public [12]. Use of crowdsensing systems by the general public span areas such that entertainment, transportation, environmental monitoring, among others [13–19]. Recently, crowdsensing systems developed under the name of *contact tracing apps* have been deployed in response of the CoronaVirus Disease 19 (COVID-19) pandemic caused by the Severe Acute Respiratory Syndrome CoronaVirus 2 (SARS-CoV-2) [3–5] . As these systems make use of consumer devices to conduct sensing on a large-scale, they circumvent costs associated with the deployment of networks with thousands of devices especially in urban areas. Fig. 1 presents the basic components of crowdsensing systems [2]. *Sensors:* They collect data either from measurable real-world variables such temperature, heart rate, pollution, objects (i.e., photos), or Human–Computer Interaction (HCI) or system processes (i.e., how much time a person logins to a website, or opens an application). The embedding of sensors for physical quantities in portable systems is possible through the research and development of tiny machines at the micrometer scale (also known as Micro-Electro Mechanical Systems (MEMS) [20]).

- *First-level integrators:* These devices collect data from sensors and they can execute basic data filtering, and perform data aggregation an analysis. Some examples of first-level integrator devices include smartphones, drones, and IoT devices.
- *Data transport:* Current crowdsensing systems use Internet or any other communication technology that provides end-to-end communication.
- *Second-level integrators:* They collect, analyze, and store data from first-level integrators. Depending on the type of system, second-level integrators may forward data to external entities or provide data analytics support to users.

Three classes of users make use of these elements to collect data through computer applications deployed at first-level (or sometimes, second-level) integrators. These users include:

- *Task organizers:* They are interested in the deployment of sensing tasks to collect data from participants.
- *Participants:* They are users who make use of first-level integrator devices which execute sensing tasks.
- *External entities:* They represent other organizations that may be interested in the data collected by a crowdsensing system.

Fig. 2 presents the various stages during data collection in crowdsensing systems and these stages include:

- *Task distribution:* In this stage, task organizers assign sensing tasks to participants by requiring them to download sensing tasks from second-level integrators.
- *Data collection:* In this stage, data is collected by executing sensing tasks at first-level integrator devices. In addition to the data collection, first-level integrators can perform data cleansing.
- *Data submission:* In this stage, first-level integrators broadcast data to second-level integrators either continuously or when contextual rules are met (i.e., reaching a specific location).

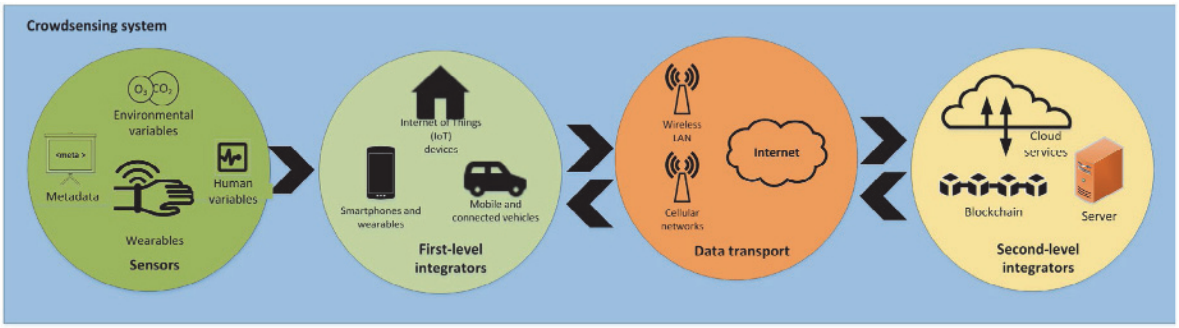

Fig. 1. Hardware components of crowdsensing systems.

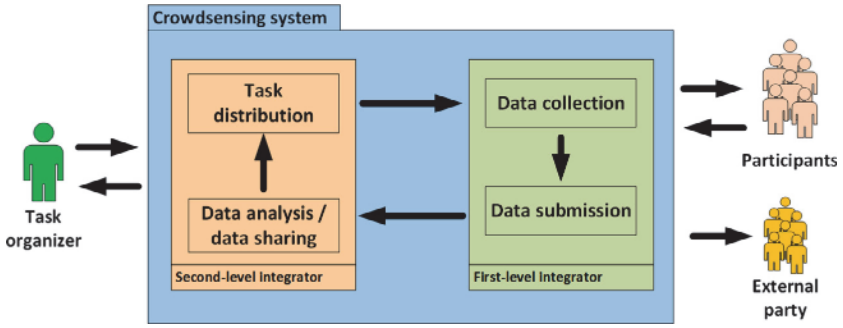

Fig. 2. Stages to collect and analyze data in crowdsensing systems [2].

- *Data analysis and sharing:* In this stage, second-level integrator devices use Artificial Intelligence (AI) and statistical methods to learn patterns from the collected data. Depending on the design of the crowdsensing system, information generated from the analysis of data may be forwarded to participants or to external entities.

We present security and privacy issues and solutions for crowdsensing systems in Tables 1 and 2. Security and privacy in crowdsensing systems involve issues at first-level and second-level integrators. From a security perspective, we can find issues related to data confidentiality, data integrity, and system availability. Some of the security issues include eavesdropping communication channels, data storage confidentiality, spoofing, authentication (for participants and sensors), exploitation of operating systems vulnerabilities, and denial of service attacks. From a privacy perspective, there are three major aspects to consider, namely, privacy issues from re-identification attacks to participants, contextual privacy issues (i.e., identifying contexts deemed private), and privacy issues when sharing data with external third parties.

2.2. Blockchain and smart contracts

Blockchain is a Peer-to-Peer (P2P) technology that implements a distributed ledger and stores data in a secure, immutable, and append-only approach through consensus or agreement among the peers in a blockchain network [66]. The structure of blockchain networks is composed of the following layers (Fig. 3) [67]:

- *A P2P network:* The P2P network ensures free communication among blockchain nodes. These blockchain nodes are around the globe and there is no hierarchical structure in the network.
- *Global distributed ledger:* The global distributed ledger implements the storage protocol to maintain the ledger. Each user is identified with a unique digital pseudonym (address) which is generated using public key cryptography. Communication between two addresses is carried out through a transaction. Data actions in the global ledger are conducted using a smart contract which execute the transactions.
  Applications: The application layer of a blockchain network implements Application Programming Interfaces (APIs) for various application scenarios. Some of these applications may include financial services, telemetry systems, copyright protection, and digital document management platforms, among others [66].

As Fig. 3 shows, the global distributed ledger consists of blocks of data chained together with cryptographic hashes. In any given block in the chain, the system stores (in all peers) transactions that are verified using a predefined set of rules to determine which transactions are valid. Only valid transactions are recorded in the blockchain. A consensus algorithm executed by all peers in the network determines the next block to be chained to the ledger [68] and it provides strong integrity to the data stored as it allows all peers to agree on a single version of the chain (guaranteeing integrity in the chain) without a central authority. Different models for consensus algorithms have been developed with various characteristics and properties. Some examples of these models include Proof-of-Work (PoF) [69,70], Proof-of-Stake (PoS) [71], Proof-of-Authority (PoA) [72], Proof-of-Space (PoSpace) [73], among others [74].

Bitcoin was created in 2008 by a person (or a group of people) under the pseudonym of Satoshi Nakamoto to develop a decentralized cryptocurrency and it is the first blockchain network publicly available [75]. Since then, blockchain technology has been extensively researched in many contexts and scenarios. Currently, blockchain networks are classified in two groups, namely public and private. In the first group (public), these networks are open to the public who can join them and execute any type of application on top of these systems. Public networks run on the Internet and common examples include Bitcoin and the

Table 1
Security issues and solutions in crowdsensing systems.

Security issues			Solutions
Data confidentiality	Eavesdropping communication channels		Encryption through protocols such as Advanced Encryption Standard (AES) and Transport Layer Security (TLS)
	Data storage confidentiality		Database encryption using various methods at different granularities (e.g., table encryption, data encryption, disk encryption)
	Privacy		Different solutions based on the privacy issue (Table 2)
Data integrity	Spoofing		Estimation and filtering (e.g., methods such as kriging and Gaussian mixture models) [21,22]
			Anomaly detection (e.g., methods such as support vector machines, neural networks, Bayesian networks) [23]
	Authentication	User authentication	Biometric methods [24,25] Smart card authentication [26,27] Two-factor authentication [28–30]
		Sensor authentication	Secure brokering hardware [31,32] Trusted execution environments [33]
System availability	Availability at first-level integrators	Interference attacks on communication between sensors and first-level integrators	Frequency hopping, sensor repositioning, protocol modification, physical layer jamming avoidance (e.g., directional antennas, spread spectrum, and channel diversity) [34]
		Battery exhaustion attacks	Power-aware operating systems [35,36] Assessing power consumption of tasks before installation [37–39] Anomaly detection for power consumption at runtime [40,41]
		Operating system vulnerabilities	Static analysis [42] Dynamic analysis [43] Formal methods [44,45]
	Availability at second-level integrators	Elasticity	Hybrids between client–server and Peer-to-Peer (P2P) architectures [11] Cloud-based solutions [46,47]
		Denial of Service (DoS)	DoS countermeasures for cloud services and traditional network environments [48]

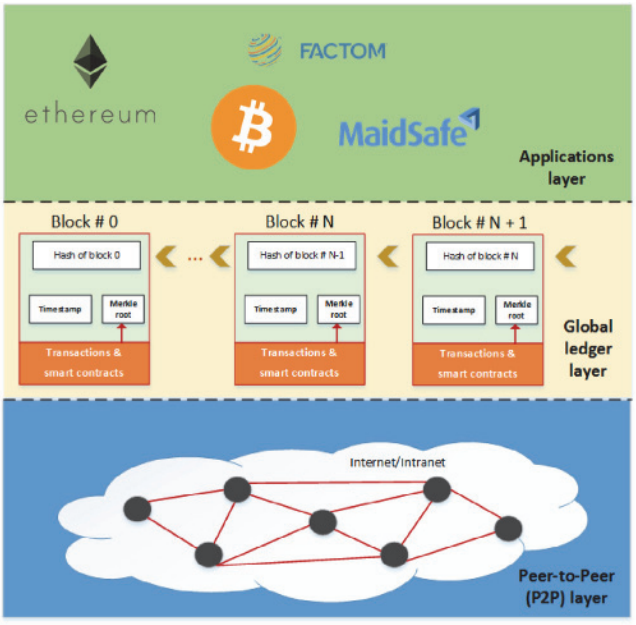

Fig. 3. Layers in a blockchain system [66].

Ethereum Mainnet [76]. In contrast, private (also called permissioned) networks provide services to private business who use these networks as part of their operations. Private networks make use of private clouds and intranets, and examples include R3 [77], and blockchain-as-a-service (BaaS) [78] platforms developed by companies such as IBM or Microsoft [79,80].

A complementary technology to Blockchain are smart contracts. If Blockchain is the technology wherein transactions are stored and maintained, then smart contracts can be described as the mechanisms to automate these transactions. Current generation of blockchain-based smart contracts require the following elements:

- *Programming language:* Smart contracts are specified by programming languages for a particular blockchain platform. The characteristics of the programming language (i.e., if it is Turing complete) determine the type of smart contracts that could be written for a particular blockchain platform. Common examples of programming languages for smart contracts in blockchain platforms include Solidity (for Ethereum) [81], as well as popular languages such as Python, C++, GOlang, and JavaScript.
- *Distributed ledger platform:* Smart contracts are stored in distributed ledgers and may store data as result of their execution in the ledgers. Ethereum is an example of a distributed ledger platform that supports smart contracts.
- *Virtual machine:* Even though smart contracts are stored in the distributed ledger their execution is conducted by a virtual machine at the edge of the network which processes the rules of the contract. The Ethereum Virtual Machine (EVM) is an example of a virtual machine for smart contracts.

Fig. 4 presents the lifecycle of a smart contract. Initially, the contract is developed using a programming language by a software

Table 2
Privacy issues and solutions in crowdsensing systems.

Privacy issues			Solutions
Re-identification	Re-identification from network identifiers		Disposable network identifiers [49] P2P anonymization [50] Double encryption (trusted broker) [51,52]
	Re-identification from task management	Authentication	Group authentication [51] Use of pseudonyms [52]
		Task distribution	Beacon-based distribution [51] Task downloading at crowded spaces [51] Anonymization network schemes [50]
		Data submission	Anonymization network schemes [50] Use of double encryption via brokers [51,52] Group-based signatures [51] k-anonymity [51] *l*-diversity [53] Use of pseudonyms [54] Micro-aggregation [55] Data aggregation [56]
Context privacy	Location privacy	Location-based queries	*k*-anonymity [57] Random noise in location submission [58] Use of known landmarks for location submission [59]
		Path privacy	Virtual fences [60] Fake locations [61] Cloaking regions [62]
	Sensor and metadata privacy		Allowing sensor data collection on task contexts only [63] Denying sensor data collection in contexts considered private [56]
External data sharing	Microdata release		k-anonymity [57] *l*-diversity [53] *t*-closeness [64]
	Statistical (summarized) data release		Differential privacy [65]

developer or entity (in this case a task manager). The contract is then published. At some point, the smart contract will be executed in a virtual machine of a node connected to the blockchain P2P network, and in this example it will be on a participant's machine. If the smart contract requires publishing of transactions back to the blockchain, it will do so. Smart contracts were first devised by Nick Szabo, who documented the idea of contract automation [82] as a way to implement legally binding paper-based contracts in computing systems. His goal was to assure data exchange with anybody who satisfied the constraints set forth by the contract [67]. Since smart contracts are executed by computers, they may be more functional than the previous generation of legally binding paper-based contracts of the past. Although the initial ideas for smart contracts were first devised in 1997, it was not until the development of distributed ledgers and their consensus mechanisms in the late 2000's and 2010's that smart contracts were implemented and deployed as envisioned by Szabo [83].

Current generation of smart contracts require blockchain technology. However, not all blockchains support smart contracts (to be as functional as possible, as envisioned by Szabo). For example, while Bitcoin is the most popular blockchain (because its application to cryptocurrency), it only supports basic smart contracts (those that exchange cryptocurrency) due to limitations in its design. In contrast, Ethereum's smart contracts can support complex operations beyond the exchange of cryptocurrency [6].

2.3. Enhancing crowdsensing systems with smart contracts

Smart contracts in public blockchains can enhance crowdsensing systems by creating automated agreements between task organizers and participants that guarantee not only the completion of a data collection task, but also automated payments for those types of crowdsensing systems that make use of monetary incentives for data collection. The data collected can also be directly stored in the blockchain itself, thus providing tamper-proof assurances that anyone can verify. We can classify the architectural models for crowdsensing systems with blockchain/smart contracts support into two categories:

- *Pure blockchain-based crowdsensing system:* In this category task organizers and participants coordinate their sensing tasks through smart contracts and blockchains. Participants execute smart contracts published in the blockchain by task organizers, data collected from participants and first-level integrators is stored in the blockchain. Task organizers download data from the blockchain and participants can be paid through cryptocurrency [84–88].
- *Hybrid models:* In these crowdsensing systems, some of the tasks (i.e., task distribution, data collection, rewards payment) are executed through smart contracts and blockchains, while others are conducted using centralized crowdsensing architectures [89–93].

The enhancement of crowdsensing systems with smart contracts offers advantages over traditional centralized crowdsensing systems in terms of incentives, data integrity, transparency, decentralization, fault tolerance, among others. The utilization of smart contracts and blockchain offers solutions to availability issues that are related to Distributed Denial of Service (DDoS) attacks, authentication, and privacy (i.e., anonymization of users without using third-parties because of their design).

3. Secure and privacy-preserving crowdsensing using smart contracts: Issues

In this section we explore issues in the development of crowdsensing systems using blockchain and smart contracts.

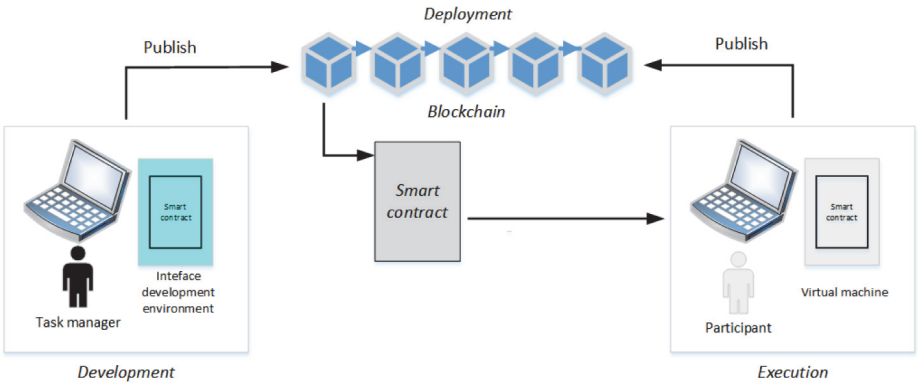

Fig. 4. Smart contract lifecycle.

3.1. Software security in smart contracts

Exploited bugs, errors and attacks in smart contracts have resulted in lost or stolen cryptocurrencies, some of them in the equivalent of millions of dollars [94]. For example, the Distributed Autonomous Organization (DAO) bug exploited a recursive call in a smart contract on the Ethereum network that forced a hard fork (a change in the protocol to invalidate blocks and transactions which require an update on the nodes in the P2P network) to claim approximately 3.6 million *ether* (an *ether* being the cryptocurrency for Ethereum) [95].

Making smart contracts more secure is a major issue for IoT and crowdsensing systems because the execution of flawed smart contracts in cyber physical systems can have a devastating potential as devices could be reprogrammed through a sensing task to steal data or create physical harm to the user [96]. Many IoT devices not only collect data but also perform some type of physical action (e.g., opening doors, increasing building temperatures, driving cars without human intervention, or delivering medication automatically to user's body).

Smart contracts could be abused to instruct IoT devices to be used as zombies by botnets to attack external parties. The Distributed Denial of Service (DDoS) attack directed at Domain Name Servers (DNS) through consumer Internet-connected cameras that disrupted the Web in 2016 is an example of these type of devastating attacks [97]. This issue is further exacerbated by the myriad of connected devices, software frameworks and service, and a lack of security by design in many of these devices' manufacturers and software/service providers.

3.2. Data integrity

For crowdsensing systems, data integrity is affected when participants submit false or misleading data for personal profit or attack a system either unintentionally or on purpose [22,98]. According to Zhang et al. [99] in a study wherein 20 participants collected barometric pressure data for seven days, they found that, not using a system to filter out spurious data led to a discrepancy of 20% from the ground truth. In crowdsensing systems over distributed ledgers this problem is exacerbated further because the sensing tasks is controlled by a smart contract that would pay the participants when data is submitted. Thus, participants may submit the same data under different identities to maximize profit [87].

3.3. Privacy

Privacy in blockchain systems has been receiving a lot of attention in recent years [66,100,101] because these systems have been developed with transparency in mind. Public blockchain systems allow (as part of their design) transactions on ledgers to be publicly checked, traced, and audited to build trust in these systems. The effect is that although transactions in ledgers are registered for users under wallets and pseudonyms, they can still be potentially re-identified [102]. In crowdsensing systems privacy is a major issue because the data collected can potentially reveal aspects considered private by the participants, making participants hesitant to participate [103,104]. Privacy leaks in crowdsensing systems may hinder the participation of the crowd, and even though the design of blockchains through wallets and pseudonyms can alleviate some privacy concerns, the potential for re-identification remains an important issue.

4. Secure and privacy-preserving crowdsensing using smart contracts: Solutions

4.1. Software security in smart contracts

Static analysis [42], dynamic analysis [43] and formal methods [44,45] for malware detection have existed before the advent of general-purpose smart contracts as tools to improve security. In static analysis the goal is to analyze the source code before execution to find possible bugs in the code [105]. A specific tool for this purpose in smart contracts is the Oyente tool [94]. Proposed by Luu et al. this tool makes use of static analysis through symbolic expressions that represent smart contract's program variables and symbolic paths [94]. Rules are then placed on the paths and if a path cannot satisfy a constraint, it is deemed infeasible. When a path is infeasible, the tool has found a possible bug with the program. In dynamic analysis the goal is to find bugs and errors through the execution of fragments of code (or equivalent transformations). Some examples of this approach in smart contracts include Manticore [106], Methryl [107], VerX [108] and KEVM [109]. In these systems smart contracts are transformed to symbolic expressions and symbolic paths which are then executed. In the third type of techniques (formal methods), the goal is to use logic and specifications to prove program correctness. Examples of formal specifications in smart contracts include the use of the F* functional programming language [110], VeriSol [111], VeriSolid [112] and SPIN [113].

A second class of solutions for software security in smart contracts relies not on static or dynamic code analysis, but on the reputation of the task managers. Systems such as SenseChain [87] CrowdBC [114] provide mechanisms in which task managers with bad behavior can be penalized through reputation so that participants do not collect data on their behalf. Similar approaches have been proposed for open mobile application markets. For example, at some point Android applications used to be published

in official markets such as Google Play without some type of vetting with limited success [115,116], which is no longer the case because of the implications of security violations and their implications before it can be signaled as such. Thus, in the context of Android devices in the Google Play Market, applications are currently being vetted before release.

4.2. Data integrity

As we have previously mentioned, data integrity is the problem wherein participants submit false or misleading data for personal profit, or to attack a system either unintentionally or on purpose. Solutions for this issue to improve the Quality of Information (QoI) have been proposed in the literature by using incentives and creating reputation measures for participants, having participants to submit some reimbursable deposits, and the utilization of trusted third-party verification [87,99,114,117].

In the first approach (incentives and reputation), the goal is to provide incentives to participants if they submit data that increases the QoI, thus improving the data integrity in the system [87]. A similar idea is the usage of reputation measures. In this approach, the goal is for task managers to keep scores for participants and allocate tasks only to those with good reputation scores [99]. In the second group (reimbursable deposits), the goal is for participants to pay a subscription to collect data. If the data satisfies integrity constraints, then the participant is paid a monetary incentive and the deposit is reimbursed. If the participant does not submit data with good quality, then the participant forfeits his/her deposit to the task manager [114]. The utilization of third-party verifiers has also been proposed. In these systems, the fundamental idea is to use a third party who verifies that the data collected by the participants satisfies data integrity constraints. In this approach, data is sent from the participant to the third party, and the third party is trusted by both the task manager and the participants [117].

4.3. Privacy

Solutions for participants' privacy protection in crowdsensing applications using smart contracts can be classified into three major groups: (1) Zero-Knowledge Proofs (ZKP); (2) external identity servers; (3) adaptations of *k*-anonymity. An example of a system that makes use of ZKP is Hawk [102] which keeps information about transactions encrypted in a blockchain, and the verification about the execution of a smart contract relies on zero-knowledge proofs. By making use of ZKP, the execution of a smart contract can be verified while keeping private data about the transaction, thus keeping users' identity private. In the second approach (external identity servers), systems use a registration server wherein participants register outside the blockchain to obtain public/private keys generated by a task organizer. These keys are then used to create an address which transactions use in the blockchain [114,118]. In the third group, adaptations of *k*-anonymity (a technique for microdata release in databases [119]) have also been proposed for blockchain and crowdsensing systems using smart contracts. In these adaptations, *k*-anonymity has been used to create *k*-anonymous groups among multiple participants who trust each other [120], and also in frameworks wherein a single participant posts his/her collected data to the blockchain under different blockchain identifiers (i.e., addresses) [121].

5. Future challenges of secure and privacy-preserving crowdsensing using smart contracts

5.1. Software engineering practices for smart contract development

When smart contracts are used in crowdsensing systems, participants may be exposed to risks arising from a lack standardization in security and privacy practices in blockchain implementations. A lack of standards can make blockchains susceptible to software bugs due to a lack of quality control in a blockchain's development process. Almost every blockchain has its own protocols, specifications, programming languages, and tools, and no standards exist to evaluate security or privacy protections in blockchain systems.

Since smart contracts are software that can make use of blockchain technologies, more research is needed in the application of software engineering practices for smart contracts and also in the development of standards among blockchain implementations to enable a minimum level of security by design models [122,123]. These practices can lead to more robust crowdsensing systems supported by blockchain and smart contracts that participants can trust.

5.2. General Data Protection Regulation (GDPR), privacy, and smart contracts

The European Union's General Data Protection Regulation (GDPR) standardized privacy requirements for online services collecting data from EU citizens independent of the services' physical location [124]. Among the privacy protections for EU citizens in GDPR is the right to erasure [125] which states that EU citizens have the right to request the deletion of any information about them that a service may have collected (this right is also often known as the right to be forgotten).

Given that some of the data collected in a crowdsensing system may be human-centric data, if a EU citizen participates in a crowdsensing system supported by a blockchain, then his/her participation in the system may be against GDPR due to blockchain's immutability, transparency, and fault-tolerance features. To address this problem, one recent solution proposed the use of smart contracts to prune a block from the blockchain if required [126], while some other solutions include hard-forks and redactable blockchains [127,128]. The latter solution allow data to be erased without hard-forks. More research on smart contracts, digital wallets, and blockchains is needed to support crowdsensing systems to satisfy legal requirements including the right to erasure, and to support other GDPR legal requirements and future privacy laws.

5.3. Scalability of blockchain-based smart contracts for crowdsensing

The current version of smart contracts is based on blockchains because blockchains provide an environment in which any external party can review a smart contract and verify the results of a smart contract's execution without the need of a central authority. However, because of consensus mechanisms in distributed ledgers, current blockchains can validate, verify, and process a small number of transactions per second (Ethereum, while it supports general-purpose smart contracts, can process approximately 15 transactions per second (around 1.3 million transactions per day) [129]), which does not scale well for the potential number of transactions that a crowdsensing system may generate.

For example, in 2018, Uber (a crowdsensing system for share riding) completed 15 million rides per day [130] (around 176

transactions per second assuming each ride is a completed transaction). For a public distributed ledger to handle that number of transactions per second for a crowdsensing system while also processing other transactions, it would need a complete overhaul of its consensus mechanisms and network architecture to be scalable [131].

In October 2021, Ethereum began updating its protocols and network topology architecture to process securely many transactions (in the order of 1000's per second) by changing its consensus mechanism from Proof of Work (PoW) to Proof of Stake (PoS) combined with a technique to spread its network load among 64 parallel chains (a technique called *shard chains*) [132]. This change reduced the power consumption of the complete Ethereum system while providing scalability. Whether this type of update will successfully support secure crowdsensing systems in public blockchains remains be seen in coming years. More work is needed to scale crowdsensing systems based on public blockchains.

5.4. Smart contracts without blockchain for crowdsensing systems

The first generation of smart contracts were developed under the idea that they need blockchain platforms because blockchains provide a tamper-proof environment to verify a smart contract's functionality, its execution, and its results without a central authority. An alternative is to explore blockchain-less smart contracts. After all, every time a user pays a service or a good using a mobile payment application (such as a mobile wallet or an NFC-enabled wearable/mobile device), the wearable/mobile device executes code in a tamper-proof environment called a Trusted Execution Environment (TEE) that proofs that a transaction executed successfully and securely. While the results of the execution of a mobile payment application are not stored in a decentralized ledger, the results of the transaction can still be verified by a third-party (a bank) and they are legally enforceable. For crowdsensing systems, the use of blockchain-based smart contracts opens up decentralized data markets wherein participants could sell sensor data and be paid in cryptocurrency [6]. However, if a fiat currency [133] is used to pay as incentives to participate in data markets, blockchains may not be needed after all. The development of a specification for blockchain-less smart contracts remains a topic of further research.

6. Conclusion

The dawn of Blockchain, cryptocurrencies, and smart contracts in the last few years has led to the emergence of exciting applications. Crowdsensing systems is one type of applications that can benefit from the utilization of smart contracts and blockchain systems. Security and privacy are important aspects for these systems. We highlighted some of these issues along with possible solutions. Finally, we have identified future research challenges that must be addressed in the future to deploy secure and privacy-preserving crowdsensing systems that take advantage of smart contracts and blockchain technology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the anonymous reviewers for their valuable comments which helped us improve the content, quality, and presentation of this paper.

Funding

This research was supported by the U.S. National Science Foundation under grant award no. 1950416.

References

[1] A.J. Perez, S. Zeadally, Recent advances in wearable sensing technologies, Sensors. 21 (20) (2021) 6828.
[2] A.J. Perez, S. Zeadally, Design and evaluation of a privacy architecture for crowdsensing applications, ACM SIGAPP Appl. Comput. Rev. 18 (1) (2018) 7–18.
[3] D.A. Drew, L.H. Nguyen, C.J. Steves, C. Menni, M. Freydin, T. Varsavsky, C.H. Sudre, M.J. Cardoso, S. Ourselin, J. Wolf, T.D. Spector, Rapid implementation of mobile technology for real-time epidemiology of COVID-19, Science (2020).
[4] H. Cho, D. Ippolito, Y.W. Yu, Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs, 2020, arXiv preprint arXiv:2003.11511.
[5] J.H. Wright, R. Caudill, Remote treatment delivery in response to the COVID-19 pandemic, Psychother. Psychosom. 89 (3) (2020) 1.
[6] Y. Kurt Peker, X. Rodriguez, J. Ericsson, S.J. Lee, A.J. Perez, A cost analysis of internet of things sensor data storage on blockchain via smart contracts, Electronics 9 (2) (2020) 244.
[7] I. Bashir, Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained, Packt Publishing Ltd, 2018.
[8] J.S. Park, T.Y. Youn, H.B. Kim, K.H. Rhee, S.U. Shin, Smart contract-based review system for an IoT data marketplace, Sensors 18 (10) (2018) 3577.
[9] A. Javaid, N. Javaid, M. Imran, Ensuring analyzing and monetization of data using data science and blockchain in lot devices, (Doctoral dissertation, MS thesis), COMSATS University Islamabad (CUI), Islamabad 44000, Pakistan, 2019.
[10] C.Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proc. IEEE 91 (8) (2003) 1247–1256.
[11] A.J. Perez, M.A. Labrador, S.J. Barbeau, G-sense: a scalable architecture for global sensing and monitoring, IEEE Netw. 24 (4) (2010) 57–64.
[12] A.T. Campbell, S.B. Eisenman, N.D. Lane, E. Miluzzo, R.A. Peterson, People-centric urban sensing, in: Proceedings of the 2nd Annual International Workshop on Wireless Internet, 2006, p. 18.
[13] E. Kanjo, Noisespy: A real-time mobile phone platform for urban noise monitoring and mapping, Mob. Netw. Appl. 15 (4) (2010) 562–574.
[14] W.Z. Khan, Y. Xiang, M.Y. Aalsalem, Q. Arshad, Mobile phone sensing systems: A survey, IEEE Commun. Surv. Tutor. 15 (1) (2012) 402–427.
[15] N.D. Lane, S.B. Eisenman, M. Musolesi, E. Miluzzo, A.T. Campbell, Urban sensing systems: opportunistic or participatory? in: Proceedings of the 9th Workshop on Mobile Computing Systems and Applications, 2008, pp. 11–16, http://dx.doi.org/10.1145/1411759.1411763.
[16] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A.T. Campbell, A survey of mobile phone sensing, IEEE Commun. Mag. 48 (9) (2010) http://dx.doi.org/10.1109/MCOM.2010.5560598.
[17] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, L. Selavo, Real time pothole detection using android smartphones with accelerometers, in: Proceedings of 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011, http://dx.doi.org/10.1109/DCOSS.2011.5982206.
[18] D. Mendez, A.J. Perez, M.A. Labrador, J.J. Marron, P-sense: A participatory sensing system for air pol-lution monitoring and control, in: Proceedings of the 2011 IEEE International Conference on Pervasive Computing and Communications (PERCOM), 2011, pp. 344–347, http://dx.doi.org/10.1109/PERCOMW.2011.5766902.
[19] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, P. Boda, PEIR, the personal environmental impact report, as a plat-form for participatory sensing systems research, in: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09), 2009, pp. 55–68, http://dx.doi.org/10.1145/1555816.1555823.
[20] M. Gadel-Hak (Ed.), The MEMS handbook, CRC Press, 2001.
[21] D. Mendez, M. Labrador, K. Ramachandran, Data interpolation for participatory sensing systems, Pervasive Mob. Comput. 9 (1) (2013) 132–148.
[22] D. Mendez, M.A. Labrador, On sensor data verification for participatory sensing systems, J. Netw. 8 (3) (2013) 576.
[23] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 1–58.
[24] C.C. Poon, Y.T. Zhang, S.D. Bao, A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health, IEEE Commun. Mag. 44 (4) (2006) 73–81.

[25] N. Henry, N. Paul, N. McFarlane, Using bowel sounds to create a forensically-aware insulin pump system, in: 2013 USENIX Workshop on Health Information Technologies (HealthTech 13), 2013.
[26] O. Mir, T.van.der. Weide, C.C. Lee, A secure user anonymity and authentication scheme using AVISPA for telecare medical information systems, J. Med. Syst. 39 (9) (2015) 89.
[27] J.M. Sorber, M. Shin, R. Peterson, D. Kotz, Plug-n-trust: practical trusted sensing for mhealth, in: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, 2012, pp. 309-322.
[28] L. Xu, F. Wu, Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care, J. Med. Syst. 39 (2) (2015) 10.
[29] V.T. Wilder, Y. Gao, S.P. Wang, A.J. Perez, Multi-factor stateful authentication using NFC, and mobile phones, in: 2019 SoutheastCon, IEEE, 2019, pp. 1–6.
[30] F. Wu, L. Xu, S. Kumari, X. Li, An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks, Multimedia Syst. 23 (2) (2017) 195–205.
[31] V. Pournaghshband, M. Sarrafzadeh, P. Reiher, Securing legacy mobile medical devices, in: International Conference on Wireless Mobile Communication and Healthcare, Springer, Berlin, Heidelberg, 2012, pp. 163–172.
[32] A. Darwish, A.E. Hassanien, Wearable and implantable wireless sensor network solutions for healthcare monitoring, Sensors 11 (6) (2011) 5561–5595.
[33] J.E. Ekberg, K. Kostiainen, N. Asokan, The untapped potential of trusted execution environments on mobile devices, IEEE Secur. Privacy 12 (4) (2014) 29–37.
[34] K. Pelechrinis, M. Iliofotou, S.V. Krishnamurthy, Denial of service attacks in wireless networks: The case of jammers, IEEE Commun. Surv. Tutor. 13 (2) (2010) 245–257.
[35] N. Vallina-Rodriguez, J. Crowcroft, ErdOS: achieving energy savings in mobile OS, in: Proceedings of the Sixth International Workshop on MobiArch, 2011, pp. 37-42.
[36] A. Merlo, M. Migliardi, L. Caviglione, A survey on energy-aware security mechanisms, Pervasive Mob. Comput. 24 (2015) 77–90.
[37] M. Dong, L. Zhong, Self-constructive high-rate system energy modeling for battery-powered mobile systems, in: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, 2011, pp. 335-348.
[38] C. Min, Y. Lee, C. Yoo, S. Kang, S. Choi, P. Park, I. Hwang, Y. Ju, S. Choi, J. Song, PowerForecaster: Predicting smartphone power impact of continuous sensing applications at pre-installation time, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, 2015, pp. 31-44.
[39] R. Mittal, A. Kansal, R. Chandra, Empowering developers to estimate app energy consumption, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, 2012, pp. 317-328.
[40] X. Ma, P. Huang, X. Jin, P. Wang, S. Park, D. Shen, Y. Zhou, L.K. Saul, G.M. Voelker, eDoctor: automatically diagnosing abnormal battery drain issues on smartphones, in: Proceedings of NSDI 2013, 2013.
[41] F. Xu, Y. Liu, Q. Li, Y. Zhang, V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics, in: Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13), 2013, pp. 43-55.
[42] S. Hallem, B. Chelf, Y. Xie, D. Engler, A system and language for building system-specific, static analyses, in: Proceedings of the ACM SIGPLAN 2002 Conference on Programming Language Design and Implementation, 2002, pp. 69-82.
[43] A. Fattori, R. Paleari, L. Martignoni, M. Monga, Dynamic and transparent analysis of commodity production systems, in: Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, 2010, pp. 417-426.
[44] G.J. Holzmann, The model checker SPIN, IEEE Trans. Softw. Eng. 23 (5) (1997) 279–295.
[45] S. Gritzalis, D. Spinellis, P. Georgiadis, Security protocols over open networks and distributed systems: Formal methods for their analysis, design, and verification, Comput. Commun. 22 (8) (1999) 697–709.
[46] K. Lee, D. Murray, D. Hughes, W. Joosen, Extending sensor networks into the cloud using amazon web services, in: 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications, IEEE, 2010, pp. 1–7.
[47] Y. Xu, A. Helal, Scalable cloud–sensor architecture for the Internet of Things, IEEE Internet Things J. 3 (3) (2015) 285–298.
[48] O. Osanaiye, K.K.R. Choo, M. Dlodlo, Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework, J. Netw. Comput. Appl. 67 (2016) 147–165.
[49] M. Gruteser, D. Grunwald, Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis, Mob. Netw. Appl. 10 (3) (2005) 315–325.
[50] J. Al-Muhtadi, R. Campbell, A. Kapadia, M.D. Mickunas, S. Yi, Routing through the mist: Privacy preserving communication in ubiquitous computing environments, in: Proceedings 22nd International Conference on Distributed Computing Systems, IEEE, 2002, pp. 74–83.
[51] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, N. Triandopoulos, Anonysense: A system for anonymous opportunistic sensing, Pervasive Mob. Comput. 7 (1) (2011) 16–30.
[52] I.J. Vergara-Laurens, D. Mendez, M.A. Labrador, Privacy, quality of information, and energy consumption in participatory sensing systems, in: 2014 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2014, pp. 199–207.
[53] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, *l*-Diversity: Privacy beyond *k*-anonymity, ACM Trans. Knowl. Discov. Data (TKDD) 1 (1) (2007) 3–es.
[54] H. Lu, N. Lane, S. Eisenman, A. Campbell, Bubble-sensing: A new paradigm for binding a sensing task to the physical world using mobile phones, in: Workshop on Mobile Devices and Urban Sensing, IPSN, Vol. 8, 2008.
[55] D. Christin, P.S. López, A. Reinhardt, M. Hollick, M. Kauer, Share with strangers: Privacy bubbles as user-centered privacy control for mobile content sharing applications, information security technical report, 17, (3) 2013, pp. 105–116.
[56] A. Kapadia, T. Henderson, J.J. Fielding, D. Kotz, Virtual walls: Protecting digital privacy in pervasive environments, in: International Conference on Pervasive Computing, Springer, Berlin, Heidelberg, 2007, pp. 162–179.
[57] M. Gruteser, D. Grunwald, Anonymous usage of location-based services through spatial and temporal cloaking, in: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, 2003, pp. 31-42.
[58] P. Wightman, W. Coronell, D. Jabba, M. Jimeno, M. Labrador, Evaluation of location obfuscation techniques for privacy in location based information systems, in: 2011 IEEE Third Latin-American Conference on Communications, IEEE, 2011, pp. 1–6.
[59] J. Krumm, Inference attacks on location tracks, in: International Conference on Pervasive Computing, Springer, Berlin, Heidelberg, 2007, pp. 127–143.
[60] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.C. Herrera, A.M. Bayen, M. Annavaram, Q. Jacobson, Virtual trip lines for distributed privacy-preserving traffic monitoring, in: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, 2008, pp. 15-28.
[61] K.C. Lee, W.C. Lee, H.V. Leong, B. Zheng, Navigational path privacy protection: navigational path privacy protection, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009, pp. 691-700.
[62] K.T. Yang, G.M. Chiu, H.J. Lyu, D.J. Huang, W.C. Teng, Path privacy protection in continuous location-based services over road networks, in: 2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2012, pp. 435–442.
[63] H. Lu, N. Lane, S. Eisenman, A. Campbell, Bubble-sensing: A new paradigm for binding a sensing task to the physical world using mobile phones, in: Workshop on Mobile Devices and Urban Sensing, IPSN Vol. 8, 2008.
[64] N. Li, T. Li, S. Venkatasubramanian, T-closeness: Privacy beyond *k*-anonymity and *l*-diversity, in: 2007 IEEE 23rd International Conference on Data Engineering, IEEE, 2007, pp. 106–115.
[65] C. Dwork, Differential privacy: A survey of results, in: International Conference on Theory and Applications of Models of Computation, Springer, Berlin, Heidelberg, 2008, pp. 1–19.
[66] Q. Feng, D. He, S. Zeadally, M.K. Khan, N. Kumar, A survey on privacy protection in blockchain system, J. Netw. Comput. Appl. 126 (2019) 45–58.
[67] G. Bello, A.J. Perez, On the application of financial security standards in blockchain platforms, in: Blockchain Cybersecurity, Trust and Privacy, Springer, Cham, 2020, pp. 247–267.
[68] G.T. Nguyen, K. Kim, A survey about consensus algorithms used in blockchain, J. Inf. Process. Syst. 14 (1) (2018).
[69] C. Dwork, M. Naor, Pricing via processing or combatting junk mail, in: Annual International Cryptology Conference, Springer, Berlin, Heidelberg, 1992, pp. 139–147.
[70] M. Jakobsson, A. Juels, Proofs of work and bread pudding protocols, in: Secure Information Networks, Springer, Boston, MA, 1999, pp. 258–272.
[71] S. King, S. Nadal, Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, 2012, self-published paper, August, 19, p.1. Available online: https://www.chainwhy.com/upload/default/20180619/126a057fef926dc286accb372da46955.pdf . Accessed on December 7/2020.
[72] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain, 2018.

[73] G. Ateniese, I. Bonacina, A. Faonio, N. Galesi, Proofs of space: When space is of the essence, in: International Conference on Security and Cryptography for Networks, Springer, Cham, 2014, pp. 538–557.

[74] L.S. Sankar, M. Sindhu, M. Sethumadhavan, Survey of consensus protocols on blockchain applications, in: 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 2017, pp. 1–5.

[75] Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, Available online: https://bitcoin.org/bitcoin.pdf . Accessed on December 7 / 2020.

[76] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, in: Ethereum Project Yellow Paper, Vol. 151, 2014, pp. 1–32.

[77] Y. Guo, C. Liang, Blockchain application and outlook in the banking industry, Financial Innov. 2 (1) (2016) 24.

[78] J. Singh, J.D. Michels, Blockchain as a service (baas): providers and trust, in: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2018, pp. 67–74.

[79] P.S. Aithal, Blockchain Based Service: A Case Study on IBM Blockchain Services & Hyperledger Fabric, 2020.

[80] Y. Wang, S.K. Lahiri, S. Chen, R. Pan, I. Dillig, C. Born, I. Naseer, K. Ferles, Formal verification of workflow policies for smart contracts in azure blockchain, in: Working Conference on Verified Software: Theories, Tools, and Experiments, Springer, Cham, 2019, pp. 87–106.

[81] C. Dannen, Introducing Ethereum and Solidity, Vol. 1, A Press, Berkeley, 2017.

[82] N. Szabo, The idea of smart contracts, 1997, Available online: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html . Accessed on December 12 / 2020.

[83] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F.Y. Wang, Blockchain-enabled smart contracts: architecture, applications, and future trends, IEEE Trans. Syst. Man Cybern. 49 (11) (2019) 2266–2277.

[84] D. Chatzopoulos, S. Gujar, B. Faltings, P. Hui, Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain, in: 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), IEEE, 2018, pp. 442–450.

[85] J. Huang, L. Kong, L. Kong, Z. Liu, Z. Liu, G. Chen, Blockchain-based crowd-sensing system, in: 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), IEEE, 2018, pp. 234–235.

[86] S. Zhu, Z. Cai, H. Hu, Y. Li, W. Li, Zkcrowd: a hybrid blockchain-based crowdsourcing platform, IEEE Trans. Ind. Inf. 16 (6) (2019) 4196–4205.

[87] M. Kadadha, H. Otrok, R. Mizouni, S. Singh, A. Ouali, Sensechain: A blockchain-based crowdsensing framework for multiple requesters and multiple workers, Future Gener. Comput. Syst. 105 (2020) 650–664.

[88] Y. Lu, Q. Tang, G. Wang, Zebralancer: Private and anonymous crowd-sourcing system atop open blockchain, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2018, pp. 853–865.

[89] S. Delgado-Segura, C. Tanas, J. Herrera-Joancomartí, Reputation and reward: Two sides of the same bitcoin, Sensors 16 (6) (2016) 776.

[90] J. Kang, Z. Xiong, D. Niyato, S. Xie, J. Zhang, Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory, IEEE Internet Things J. 6 (6) (2019) 10700–10714.

[91] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things, Ieee Access 4 (2016) 2292–2303.

[92] P. Bellavista, M. Cilloni, G.Di. Modica, R. Montanari, P.C.M. Picone, M. Solimando, An edge-based distributed ledger architecture for supporting decentralized incentives in mobile crowdsensing, in: 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), IEEE, 2020, pp. 781–787.

[93] L. Wei, J. Wu, A blockchain-based hybrid incentive model for crowdsensing, Electronics 9 (2) (2020) 215.

[94] L. Luu, D.H. Chu, H. Olickel, P. Saxena, A. Hobor, Making smart contracts smarter, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 254-269.

[95] O.G. Güçlütürk, The DAO hack explained: Unfortunate take-off of smart contracts, Medium (2018).

[96] CBSNews, 'Pokemon Go' being used to stage robberies, police say, Available at: http://www.cbsnews.com/news/robbery-suspects-using-pokemon-go-to-target-victims-police-say/. Accessed December 7, 2020.

[97] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J.A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, Understanding the mirai botnet, in: 26th USENIX security symposium (USENIX Security 17), 2017, pp. 1093-1110.

[98] D. He, S. Chan, M. Guizani, User privacy and data trustworthiness in mobile crowd sensing, IEEE Wirel. Commun. 22 (1) (2015) 28–34.

[99] H. Zhang, S. Bagchi, H. Wang, Integrity of data in a mobile crowdsensing campaign: A case study, in: Proceedings of the First ACM Workshop on Mobile Crowdsensing Systems and Applications, 2017, pp. 50-55.

[100] Q. Feng, D. He, S. Zeadally, K. Liang, BPAS: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks, IEEE Trans. Ind. Inf. 16 (6) (2019) 4146–4155.

[101] C. Lin, D. He, S. Zeadally, N. Kumar, K.K.R. Choo, SecBCS: a secure and privacy-preserving blockchain-based crowdsourcing system, Sci. China Inf. Sci. 63 (3) (2020) 1–14.

[102] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in: 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 839–858.

[103] A.J. Perez, S. Zeadally, PEAR: A privacy-enabled architecture for crowd-sensing, in: Proceedings of the International Conference on Research in Adaptive and Convergent Systems, 2017, pp. 166-171.

[104] A.J. Perez, S. Zeadally, N. Jabeur, Security and privacy in ubiquitous sensor networks, J. Inf. Process. Syst. 14 (2) (2018).

[105] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, Y. Alexandrov, Smartcheck: Static analysis of ethereum smart contracts, in: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, 2018, pp. 9-16.

[106] M. Mossberg, F. Manzano, E. Hennenfent, A. Groce, G. Grieco, J. Feist, T. Brunson, A. Dinaburg, Manticore: A user-friendly symbolic execution framework for binaries and smart contracts, in: 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2019, pp. 1186–1189.

[107] Mythril Classic, 2017, Available online: https://github.com/ConsenSys/mythril-classic. Accessed on December 7 / 2020.

[108] A. Permenev, D. Dimitrov, P. Tsankov, D. Drachsler-Cohen, M. Vechev, Verx: Safety verification of smart contracts, in: 2020 IEEE Symposium on Security and Privacy, SP, 2020, pp. 18-20.

[109] E. Hildenbrandt, M. Saxena, X. Zhu, N. Rodrigues, P. Daian, D. Guth, G. Rosu, Kevm: A complete semantics of the ethereum virtual machine, 2017.

[110] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, S. Zanella-Béguelin, Formal verification of smart contracts: Short paper, in: Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, 2016, pp. 91-96.

[111] Y. Wang, S.K. Lahiri, S. Chen, R. Pan, I. Dillig, C. Born, I. Naseer, Formal specification and verification of smart contracts for azure blockchain, 2018, arXiv preprint arXiv:1812.08829.

[112] A. Mavridou, A. Laszka, E. Stachtiari, A. Dubey, Verisolid: Correct-by-design smart contracts for ethereum, in: International Conference on Financial Cryptography and Data Security, Springer, Cham, 2019, pp. 446–465.

[113] X. Bai, Z. Cheng, Z. Duan, K. Hu, Formal modeling and verification of smart contracts, in: Proceedings of the 2018 7th International Conference on Software and Computer Applications, 201, pp. 322-326.

[114] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.N. Liu, Y. Xiang, R.H. Deng, Crowdbc: A blockchain-based decentralized framework for crowdsourcing, IEEE Trans. Parallel Distrib. Syst. 30 (6) (2018) 1251–1266.

[115] N. Viennot, E. Garcia, J. Nieh, A measurement study of google play, in: The 2014 ACM International Conference on Measurement and Modeling of Computer Systems, 2014, pp. 221-233.

[116] A. Mylonas, A. Kastania, D. Gritzalis, Delegate the smartphone user? Security awareness in smartphone platforms, Comput. Secur. 34 (2013) 47–66.

[117] D. Liang, J. An, J. Cheng, H. Yang, R. Gui, The quality control in crowdsensing based on twice consensuses of blockchain, in: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, 2018, pp. 630-635.

[118] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, W. Dou, A blockchain-powered crowdsourcing method with privacy preservation in mobile environment, IEEE Trans. Comput. Soc. Syst. 6 (6) (2019) 1407–1419.

[119] P. Samarati, Protecting respondents' identities in microdata release, IEEE Trans. Knowl. Data Eng. 13 (6) (2001) 1010–1027.

[120] J. Wang, M. Li, Y. He, H. Li, K. Xiao, C. Wang, A blockchain based privacy-preserving incentive mechanism in crowdsensing applications, IEEE Access 6 (2018) 17545–17556.

[121] M. Ober, S. Katzenbeisser, K. Hamacher, Structure and anonymity of the bitcoin transaction graph, Future Internet 5 (2) (2013) 237–250.

[122] S. Porru, A. Pinna, M. Marchesi, R. Tonelli, Blockchain-oriented software engineering: challenges and new directions, in: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), IEEE, 2017, pp. 169–171.

[123] G. Destefanis, M. Marchesi, M. Ortu, R. Tonelli, A. Bracciali, R. Hierons, Smart contracts vulnerabilities: a call for blockchain software engineering? in: 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), IEEE, 2018, pp. 19–25.

[124] M. Goddard, The EU general data protection regulation (GDPR): European regulation that has a global impact, Int. J. Mark. Res. 59 (2017) 703–705.

[125] European Union, A guide to GDPR data privacy requirements, 2020, Available online https://gdpr.eu/data-privacy. Accessed on November 1 / 2020.
[126] S. Farshid, A. Reitz, P. Roßbach, Design of a forgetting blockchain: A possible way to accomplish GDPR compatibility, in: Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.
[127] G. Ateniese, B. Magri, D. Venturi, E. Andrade, Redactable blockchain–or–rewriting history in bitcoin and friends, in: 2017 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2017, pp. 111–126.
[128] M. Florian, S. Henningsen, S. Beaucamp, B. Scheuermann, Erasing data from blockchain nodes, in: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2019, pp. 367–376.
[129] M. Bez, G. Fornari, T. Vardanega, The scalability challenge of ethereum: An initial quantitative analysis, in: 2019 Ieee International Conference on Service-Oriented System Engineering (Sose), IEEE, 2019, pp. 167–176.
[130] D. Wigan, Uber global wealth chains, in: Combating Fiscal Fraud and Empowering Regulators, Oxford University Press, 2021, pp. 194–214.
[131] Y. Fu, C. Soman, Real-time Data Infrastructure at Uber, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 2503-2516.
[132] The Eth2 Upgrades, 2021, Available online: https://ethereum.org/en/eth2/. Accessed on November 1 / 2021.
[133] B. Eichengreen, From commodity to fiat and now to crypto: what does history tell us?, No. w25426, National Bureau of Economic Research, 2019, Available online: https://www.nber.org/papers/w25426. Accessed on November 1 / 2020.